In the claims:

All of the claims presented for examination are reproduced below.

1.(Currently amended) A system for providing network security by managing and manipulating ~~formed~~ live data connections and connection attempts initiated over a data-packet-network between at least two nodes connected to the network comprising:

a system host machine connected to the network;

a first software application residing on the system host machine for detecting and monitoring the live connections and connection attempts;

a data store for storing data about the live connections and connection attempts;

a second software application for emulating one or more end nodes of the connections or connection attempts; and

a third software application for detecting virus activity by hashing data passed over the live connection in real time and for comparing the hash data to a dataset containing virus signatures, the dataset searchable by hash table index, the hash entries therein derived individually from separate virus signatures;

characterized in that the system using the detection software detects one or more pre-defined states associated with a particular ~~formed~~ connection or connection attempt in progress including those associated with any data content or type transferred, ~~there over~~ and performs at least one packet generation and insertion action triggered by the detected state or states, the packet or packets emulating one or more end nodes of the connection or connection attempt to cause preemption or resolution of the detected state or states, and the hashing routine utilizes at least one sliding checksum window processing, in real time, the data passed over the live connection.

2. (Original) The system of claim 1 wherein the data-packet-network encompasses a Local Area Network connected to the Internet network enhanced with Transfer Control Protocol over Internet Protocol and User Datagram Protocol over Internet Protocol.

3. (Original) The system of claim 1 wherein the system host machine is one of a desktop computer, a router, an embedded system, a laptop computer, or a server.

4. (Original) The system of claim 1 wherein the system host is an especially dedicated piece of hardware.

5. (Original) The system of claim 1 wherein emulation of the end nodes of the connections or connection attempts is performed by generation and insertion into a data stream of the connection or connection attempt data packets using Transfer Control Protocol over Internet Protocol, the packets emulating packets from the current sending node in the connection.

6. (Original) The system of claim 5 wherein the packets inserted into a connection or connection attempt are one or a combination of Transfer Control Protocol reset packets or Transfer Control Protocol FIN packets.

7. (Original) The system of claim 1 wherein the nodes participating in the connections or connection attempts are desktop computers, servers, embedded systems, laptop computers or a combination thereof.

8. (Original) The system of claim 1 wherein the data-packet-network is an Ethernet network connected to the Internet network and the first software application is an Ethernet driver set to operate in promiscuous mode.

9. (Original) The system of claim 1 wherein the data about the connections or connection attempts includes one, more, or a combination of sender and receiver Internet Protocol addresses; Universal Resource Locators; source and destination ports; Transfer Control Protocol packet sequence numbers; Ethernet machine addresses; domain names; and

packet header details.

10. (Original) The system of claim 1 wherein the data store comprises segregated datasets representing one or more of banned Internet Protocol addresses; banned domain names; banned Universal Resource Locators; banned network ports; and virus signatures.

11. (Original) The system of claim 1 wherein the data store further includes Ethernet machine addresses associated with bitmap icons representing individual machine types.

12. (Original) The system of claim 10 wherein certain ones of the segregated datasets are built during runtime, maintained temporarily, and searchable by one of hash table indices or binary tree indices.

13. (Original) The system of claim 10 wherein certain ones of the segregated datasets are uploaded into host Random Access Memory upon booting of the host system.

14. (Canceled)

15. (Previously presented) The system of claim 1 wherein the at least one sliding checksum window processes a data string from the data in the live connection in real time comprising a first hash value computed from a set number of consecutive bytes in the window, compared to the hash table index and stored, a second hash value is then computed and compared to the hash table index when the window slides to the next consecutive byte in the data string, wherein the second hash value equals the first hash value minus the byte exiting the window plus the next consecutive byte of the data sting entering the window, thereby creating a high speed search algorithm for the connection.

16. (Previously presented) The system of claim 1 wherein upon detecting a hit for a virus signature, the second software application interrupts data stream processing of one or

more end points of the connection by sending a reset packet to stop download of the detected virus.

17. (Previously presented) A software application for manipulating one or more connection ends of a data network connection between two or more network nodes operating on a data-packet-network in response to detection of a pre-defined and undesirable state or states associated with the connection comprising:

a first portion thereof for detecting one or more states associated with the connection;

a second portion thereof for generating packets emulating packet activity of the connection; and

a third portion thereof for sending the emulated packet or packets to one or more parties of the connection;

wherein the pre-defined state or states includes one, more, or a combination of a banned Universal Resource Locator; a banned domain name; a detected virus signature; a banned port; and banned data content defined by filter; characterized in that the application uses a software communication stack to send one or more Transfer Control Protocol packets emulating in construction and sequence number a packet or packets sent by a sender end of the connection, the packet received by the receiver of the connection wherein the receiving end acknowledges the packet or packets as being a valid packet or packets received from the sender of the connection, the packet or packets sent causing pre-emption or resolution of the detected state or states.

18. (Original) The software application of claim 17 wherein the data-packet-network comprises a local-area-network enhanced with Transfer Control Protocol over Internet Protocol and User-Datagram Protocol over Internet Protocol.

19. (Original) The software application of claim 18 wherein the Local Area Network is an Ethernet network connected to an Internet network.

20. (Original) The software application of claim 17 wherein manipulation of connection ends is performed by generation of and insertion of data packets to one or more nodes of the connection using Transfer Control Protocol over Internet Protocol, the generated packets emulating sender packets in construction and sequence number.

21. (Original) The software application of claim 17 wherein the packets inserted into a connection data stream are one or a combination of Transfer Control Protocol reset packets or Transfer Control Protocol FIN packets emulating at least one sending party of the connection.

22. (Original) The software application of claim 17 wherein the software communication stack is an on-board Transfer Control Protocol over Internet Protocol communication stack.

23. (Canceled)

24. (Original) The software application of claim 17 wherein the connection end nodes are desktop computers, servers, embedded systems, laptop computers, or a combination thereof.

25. (Original) The software application of claim 17 wherein Transfer Control Protocol packets are generated and inserted according to pre-defined trigger events associated with existing states or knowledge of imminence thereof discovered during operation.

26. (Previously presented) The software application of claim 17 further including a portion thereof integrated with the first portion for detecting virus activity comprising:

a routine for hashing data in real time passed over a formed live data connection; and a routine for comparing the hash data to a dataset containing virus signatures, the dataset searchable by hash table index, the hash entries derived individually from the virus signatures.

27. (Currently Amended) The system of claim [[23]] 17 wherein the predefined state is banned content and resolution thereof includes inserting content including machine readable script by one or a sequence of TCP packets containing replacement content.

28. (Original) The software application of claim 26 wherein virus searching is supported by algorithm supporting generation and then comparison of created hash values derived from active connection data streams to hash table entries stored in a data store and to return a hit upon obtaining a match.

29. (Original) The software application of claim 26 wherein the third portion thereof is integrated with a messaging client for generating automated alerts to end nodes whose connections have been manipulated.

30. (Original) The software application of claim 26 including one or more sliding checksum windows for hashing data transferred over an active connection.

31. (Original) The software application of claim 30 wherein each checksum window processes 9 bytes of data 3-bytes at a time, each three-byte section treated as a single 24-bit number.

32.(Previously presented) The software application of claim 26 wherein the hash table is sparsely populated and wherein the hash table index thereof is bit-masked to reduce the overall size of the table and increase performance of the search.

33-55. (Canceled)